Learning Envelopes for Fault Detection and State Summarization [1]

Dennis DeCoste
Machine Learning Systems Group
Jet Propulsion Laboratory / Caltech
Pasadena, CA 91109
dennis.decoste@jpl.nasa.gov

Abstract --- This paper discusses a data mining approach for overcoming common problems with the traditional red-line limit-checking approach to fault detection and state summarization. It essentially involves learning and adapting parametric functions which provide context-sensitive bounds on historic time-series engineering data. Such bounds are suitable as dynamic plug-in replacements for static red-line values. They enable significantly earlier detection while maintaining low false alarm rates. An example will be discussed from recent onboard tests of this technology during the NASA Deep Space 1 (DS1) mission.

TABLE OF CONTENTS

1. INTRODUCTION

Autonomous fault detection in space systems typically involves comparing real-time data to predefined static "red-line" limits. For example, a fault in a heat regulator might be detected when a particular temperature gets higher than a given threshold. Such limits are popular because they are relatively easy to specify and use. But they have numerous disadvantages, which are becoming increasely significant as we move toward more autonomous spacecraft, including:

1) *late or missed alarms* --- red-lines are relatively weak (wide) bounds, detecting faults only once they become critical, and often even dangerous. Earlier detection would support a wider range of recovery procedures, including preventative maintenance that would extend mission life.

2) *false alarms* --- red-lines are traditionally made quite wide intensionally, in large part to avoid false ("nuisance") alarms. Nevertheless, such false alarms still occur routinely, sometimes resulting in mission operators eventually ignoring red-line alarms in those troublesome sensors altogether.

3) *failure to track system changes* --- predefined red-lines fail to capture changes during a mission, such as gradual spacecraft degraduation, environmental changes, and early mission "shake-out" (e.g. versus testbed performance).

Handling these problems autonomously is essential to autonomous fault diagnosis and recovery, since fault detection is the critical first step.

Addressing the above problems typically involves substantial manual effort. One common approach is to manually develop expert system rule bases or state models, to allow different red-line values to be associated with key different contexts (e.g. spacecraft operating modes). A second common approach is to continually monitor performance during the mission, manually refining and uploading new red-lines as warranted.

To address these problems without such high manual costs and in an automous manner, we have been developing data mining techniques which essentially extract red-line *functions* from mission data. We call this approach ELMER (Envelope Learning and Monitoring using Error Relaxation). For each sensor, we learn a pair of upper and lower bounding functions, called its "*envelope*". The inputs for each envelope are automatically restricted to a subset of the sensors --- those which are most relevant to determining the tightest bounds for which historic data seldom falls outside of them.

2. ENVELOPE LEARNING

Consider the task of predicting high H(y[t]) and low L(y[t]) bounding values for sensor y at each time t, based on input values from various other sensors X_i at various time lags, say t+1, t, and t-1. Note that one distinction between classic time-series prediction and our similar use here for detection tasks is that for detection it often makes considerable sense

[1] 0-7803-5846-5/00/$10.00 © 2000 IEEE

to use current (i.e. at t) and future (e.g. at t+1) values of the input sensors.

For brevity, we will denote the set of all input sensors as X and the full input set at each time t, over all those sensors and time lags as:

$$Z[t] = X[t+1], X[t], X[t-1].$$

Denote the parametric form of these bounds as follows:

$$H(y[t]) = \mathbf{f}(w_H, Z[t])$$

$$L(y[t]) = \mathbf{f}(w_L, Z[t])$$

for some suitable functional family **f,** such as sigmoidal forms popular in neural networks or even simple linear weight sums. In this paper we will assume we are given Z and **f**. Candidates can be supplied by users and futher refined using a wide variety of automated model selection and feature selection techniques explored in statistics and machine learning work.

In practice, it is not critical that Z contain only relevant quantities, since the parameter optimization process will tend to associate small parameters with useless inputs. Selecting an appropriate family **f** might seem more fundamentally critical to good prediction performance. However, practical constraints often dictate this decision as well. For example, in recent onboard experiments we were restricted to simple linear-weighted sums, due to both RAM and CPU limitations.

To handle such practical constraints, our focus has been on how to learn values for the parameter vectors w_H and w_L that lead to good results (e.g. low false alarm rates and better detectability than red-lines) even when given higher non-ideal inputs Z and function family **f.**

Bounds Estimation Techniques

We view ELMER as a collection of techniques for performing the task of *bounds estimation*, as opposed to traditional regression techniques which emphasis means estimation or general probability density estimation. We have formulated several methods for bounds estimation and have been exploring their various tradeoffs as well as comparing them to traditional techniques. Common to all our techniques is the notion that bounds estimation's key distinction is that it involves a special form of constrained optimization. In particular, a prediction from a learned high bounding function should not only be as close to the training target value as possible, but also strictly above that target.

Our envelope learning process is essentially a generalization of standard least squared regression, in which constraints to ensure that most data falls within the resulting bounds are enforced. There are a variety of ways to do so. The various techniques we have implemented and explored fall into two broad categories:

Memory-Based Methods --- In classic k-nearest neighbors regression [1], an estimation of the input-conditional mean for y[t] is given by averaging the values $y[t_i]$ associated with the k training examples $Z[t_i]$ "closest" to the new test example Z[t], based on some distance metric (often Euclidian) and value of k. This technique can be used to estimate high (low) bounds instead of means by essentially taking the maximum (minimum) of the $y[t_i]$ values, instead of averaging them.

Model-Based Methods --- A common "error bars" [6] approach involves estimators of input-conditional means and variances (see Figure 1). Consider a variant, where the input-conditional mean estimation is used to divide the training set into "target is above the mean" and "target is below the mean" subsets. A prediction model of the error residual for each subset is learned, allowing error distributions which are asymmetric around the mean to be easily handled. The final learned high bounding function is H(y[t]) = Hb + Hs * M(Z[t]), where M(Z[t]) is the mean estimation, Hb is a bias shift value and Hs is a scaling factor playing a role similar to a standard deviation factor in a Gaussian model.

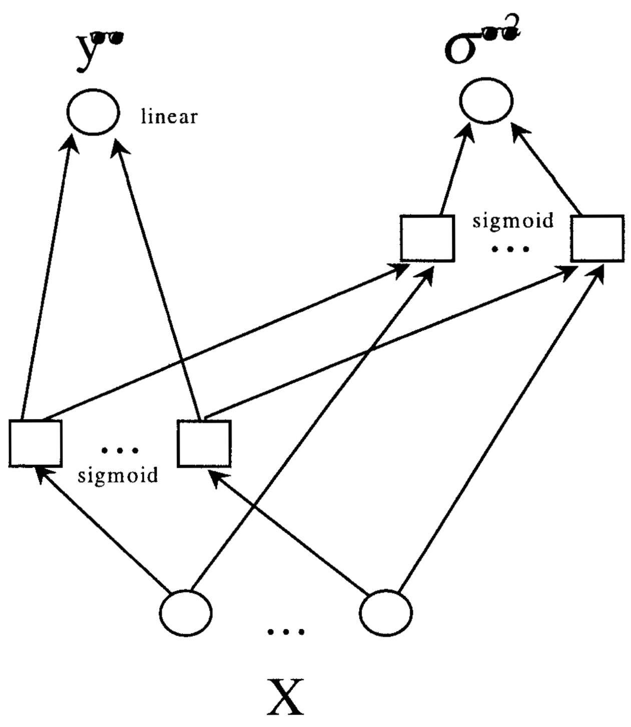

Figure 1 Network for Input-Conditional Mean and Variance

Generalizing to Future Test Data

Making such methods generalize well to future data requires further details beyond the basic sketches given above. For example, one way to determine reasonable values for the Hs and Hb parameters above is to use a simplex method of constrained optimization (over only those two variables, for fixed M(Z[t])) and extensively use cross-validation to select the widest fit (i.e. highest Hb and Hs values) that any 90% subset of the training data requires to avoid alarms on 10% hold out sets. Similarly, for methods such as min/max k-nearest neighbors, the best k value and suitable shift offsets to bound all data can be determined via cross-validation as well.

Tradeoffs Among Alternatives

The memory-based bounding methods have some key advantages, including often being more readily understood by humans during post-detection (e.g. diagnosis) analysis, since their bounds violations are grounded in terms of specific previous sensor behavior examples. We have focused more to date on model-based methods, mainly because onboard applications, including the DS1 Beacon experiment discussed later below, have tight space constraints that preclude on-line access to vast historic databases. However, we are beginning to explore, within our bounding context, appropriate ways to combine both approaches. One such approach involves support vector machines [9], which identify subsets of examples which are most valuable for retaining in memory.

An important property of ELMER is that it is very scaleable with respect to the number of available sensor inputs, the available training data, the computational time available for learning and adaptation, and the real-time memory and CPU restrictions for representing and computing final bounding functions. It finds the best bounds it can with whatever it is given (even if that results in almost static red-line bounds at that point), and can incrementally improve bounds as more is given later.

3. RELATED WORK

It is useful to view ELMER as a generalization of the static red-lines traditionally used in NASA fault monitoring operations. ELMER's bounds are intended to work just like red-lines in that data outside of those bounds should be suspect. A key problem with red-lines is that attempts to avoid "nuisance alarms", where red-lines are excessively tight, easily leads to red-lines that are much too wide to detect faults until very late (and often critical) stages. Indeed, our work on ELMER arose from attempting to better capture the context-sensitivity of the domains, for earlier detectability, while not making the strong error distribution assumptions that common statistical error bars approaches do. Note that data outside of error bars based on the mean plus or minus two standard deviations will still occur about 5% of the time. That is not acceptable for large-scale monitoring tasks, for which thousands of sensors are sampled every second.

The fundamental problem is that for complex engineering systems such as spacecraft, the error in achievable predictions based on available sensor data is not primarily Gaussian, nor any other kind of standard distribution. Even when the sensor data is sufficient to find a deterministic (plus small Gaussian white noise) model, from a practical point of view that does not help much if the step-wise regression technique being used has not yet selected all the righnputs, out of the thousands of (raw and transformed) candidates to consider. Furthermore, the mission's memory and CPU limitations might well require limiting each function to a handful of the relevant inputs.

Asymmetric Error

A key distinction between ELMER and other machine learning technologies is that it learns and defines its high and low bounds independently and without assuming a specific prediction error distribution. Other techniques, such as neural networks which learn "error bars" (e.g. estimates of the mean of the data as well as the variance of the data [6]), assume specific types of distribution of error, often symmetric (e.g. Gaussian). ELMER handles well such asymmetric error distributions, which are common in spacecraft data (due to engineering set-points and other skewed behavior). The end result is that ELMER can produce tighter bounds which lead to better detections and trending predictions.

Probability Density Estimation

There does exist a class of techniques, called probability density estimation (PDE) (e.g. [8]), which, like ELMER, avoid the problems of assuming any specific class of error distributions. Conditional probability densities explicitly represent the probability of each possible output value, given the inputs. For example, instead of assuming that error is distributed as a single Gaussian, a PDE approach such as mixture density estimation might use hundreds of Gaussians of varying parameters (i.e. centers and widths). With sufficiently large mixtures, any distribution can eventually be modeled to any arbitrary precision using such PDE.

However, the generality of PDE is both its strength and its major weakness. To learn the parameters of the mixtures well typically requires orders of magnitude more data than the single regression that ELMER requires for each bound. Similarly, PDE's with hundreds of Gaussians are orders of magnitude more expensive to store and compute at execution time, making them much more expensive than ELMER to use for tasks such as real-time monitoring.

In short, PDE promises more than is necessary for tasks that only require bounds, and delivering on those promises requires excessive resources at both training and execution. Thus, we argue that ELMER's explicit focus on estimating bounds is more appropriate for many tasks, such as monitoring and resource profiling. For some tasks, most notably control, invertable models are critical. For such tasks, PDE of some precision is generally required. One planned extension for ELMER is to generalize it with PDE capability, so that in an anytime fashion it finds the best trade-off for a given task between high/low bounds and full precision PDE.

Extreme Value Theory

Quartile and extreme value theory [7] techniques have been developed within the field of statistics to help characterize high and low values without resorting to detailed probability density estimation. Extreme value work emphasizes the fact that the vast majority of examples in most data sets are not extremas, and thus models based on them will be excessively biased toward the average cases. These techniques are based on the mathematical fact that the distributions of maximum (minimum) values (e.g. annual maximum rainfall) tends to fall into a small number of classes that can be characterized by a small number of parameters that can be estimated.

Extrema value techniques are especially popular for environmental and insurance studies. For example, they are used to estimate maximum annual rainfall or the probability of flood levels exceeding a given level. In such cases, there are natural time periods over which one can compute maximums (minimums) and the data is univariate or there are only a couple of relevant input variables. They seem less applicable to our general spacecraft monitoring context.

4. EXAMPLE

For example, in a recent experiment to evaluate our envelope approach onboard the NASA Deep Space 1 (DS1) mission, the learned envelope for a battery charge temperature sensor (P-4022) correctly represented the fact that it's value had historically been within 2 degrees of related battery temperature sensors (P-4011 and P-4021) during the first few months of the mission. A later fault was then detected because the learned high bound on sensor P-4022, whose inputs were sensors P-4011 and P-4021, was violated when this previously reliable historic relation suddenly no longer held. This fault was much more subtle than what traditional red-lines on those temperature sensors would have detected.

The bounding functions (learned using the model-based approach discussed above) for these three sensors were each linear weighted sums of the other sensors at time t:

```
HI(P-4011) = 0.134188 + 1.63289 +
0.448148 * P-4021 + 0.464419 * P-4022

LO(P-4011) = -0.605297 - 2.29489 +
0.837963 * P-4021 + 0.305041 * P-4022

HI(P-4021) = 0.237627 + 1.33672 +
1.19174  * P-4011 - 0.198575 * P-4022

LO(P-4021) = -0.309171 + 1.09749 +
0.138695 * P-4011 + 0.778735 * P-4022

HI(P-4022) = 0.361257 + 1.49832 +
0.393313 * P-4011 + 0.433447 * P-4021

LO(P-4022) = -0.36943 - 2.91208 +
0.718915 * P-4011 + 0.377218 * P-4021
```

Figure 2 shows time-series plots for sensor P-4022 (and its bounds) for three data sets: training data (top), test data (middle) and test2 data (bottom). The training set represented the last 44 days of 1998. The test set represented the first 30 days of 1999, during which time the true anomaly occurred. The test2 set represented the 22 days before the training set. As expected, the training and test2 data did not get (false) alarms, but the test set did alarm just before and during the anomaly (indicated by dark black bar along bottom of the middle plot, representing times for which the data dropped below the low bound values).

Figure 3 shows the training data (and bounds) for all three sensors. Similarly, Figure 4 shows the test data and Figure 5 shows the test2 data.

5. CONCLUSIONS

Our bounds estimation techniques are also applicable to related tasks, such as "resource profiling" [4] to support planning decisions in dynamic environments (e.g. Mars Rover [5]).

Despite initial promising results, a couple of key issues must still be addressed to mature this technology for practical applications.

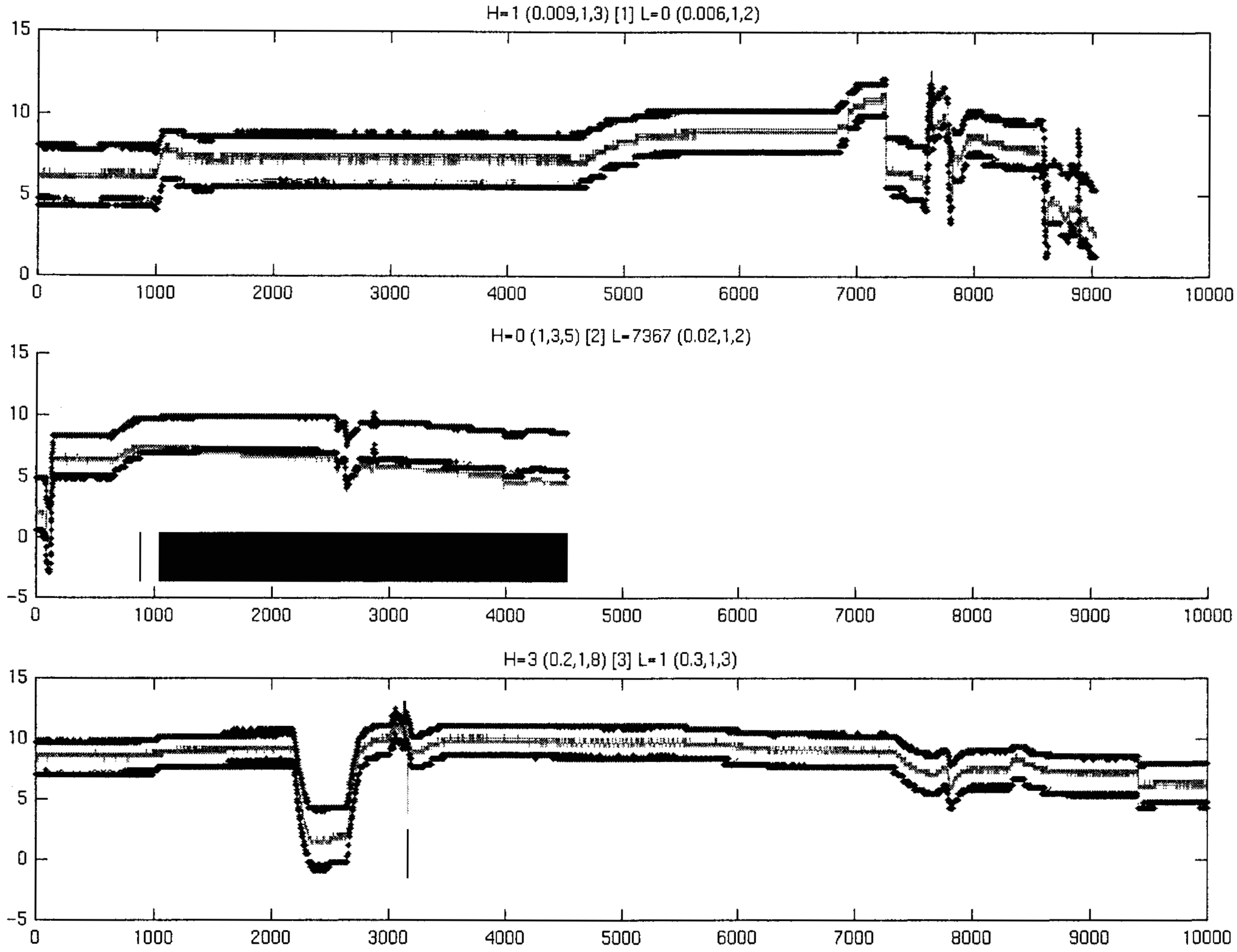

Figure 2 Sensor P-4022 for Train, Test, and Test2 Data

First, ELMER needs to be extended to allow it to determine at runtime when the current (test) data is so dissimilar from the training data that the previously learned bounds are not applicable. For example, autonomously detecting such situations is required to avoid false alarms when the test context is radically different from the training scenarios --- such as training during cruise phase of a mission and testing during orbit-insertion phase. In a general probability density estimation approach, such determination could be directly performed by evaluating some (previously learned) joint density estimate for the current values of all the input sensors. That is, a small likelihood in the conditional probability of some quantity should not itself be the cause for declaring a fault detection when the inputs for its estimation are in fact themselves very unlikely. Capturing this distinction between inputs joint probability and output conditional probability sufficiently for the goals of bounds estimation, without incurring the full cost of general density estimation, is the goal for this extension. An advantage of memory-based approaches to bounds estimation mentioned earlier is that their use of distance metrics between test and training data already provides some such distinctions (i.e. a nearest-neighbor which is still relatively far away could be an indication that the training data is insufficient to confidently bound the new data).

Second, ELMER needs to be extended to support robust on-line adaptation of bounding functions in light of new data during a mission. This capability is required to track non-stationarities due to system drift and degradation, as well when environmental conditions turn out to be different that initial expectations (e.g. ground testbed for Mars rover). To date, our work has focused on learning envelopes using batch training, due to its simplicity of implementation and evaluation. Also, addressing the issues of when to adapt and what portions of a model to retain requires first addressing the above issue of detecting significant differences between previous (training) and new (test) data.

Another goal for the ELMER work is to more formally incorporate an ability to learn probabilistic graphical models (e.g. Bayesian networks) from the data. ELMER currently uses basic concepts from the field of Bayes nets, such as partial correlations, to (heuristically) identify useful inputs for each bounding function. Useful extensions would include refining Bayes net algorithms for learning causal (directed) structures so that they work well within the bounds estimation framework of ELMER.

Finally, near term goals include more comprehensive evaluation of our various bounds estimation techniques, across several data sets, including deep space missions, Mars Rover, and Space Shuttle.

6. ACKNOWLEDGEMENTS

John Szijjarto provided significant feedback and assistance in obtaining and evaluating the DS1 data sets.

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES

[1] Chris Bishop. *Neural Networks and Pattern Recognition*, 1995.

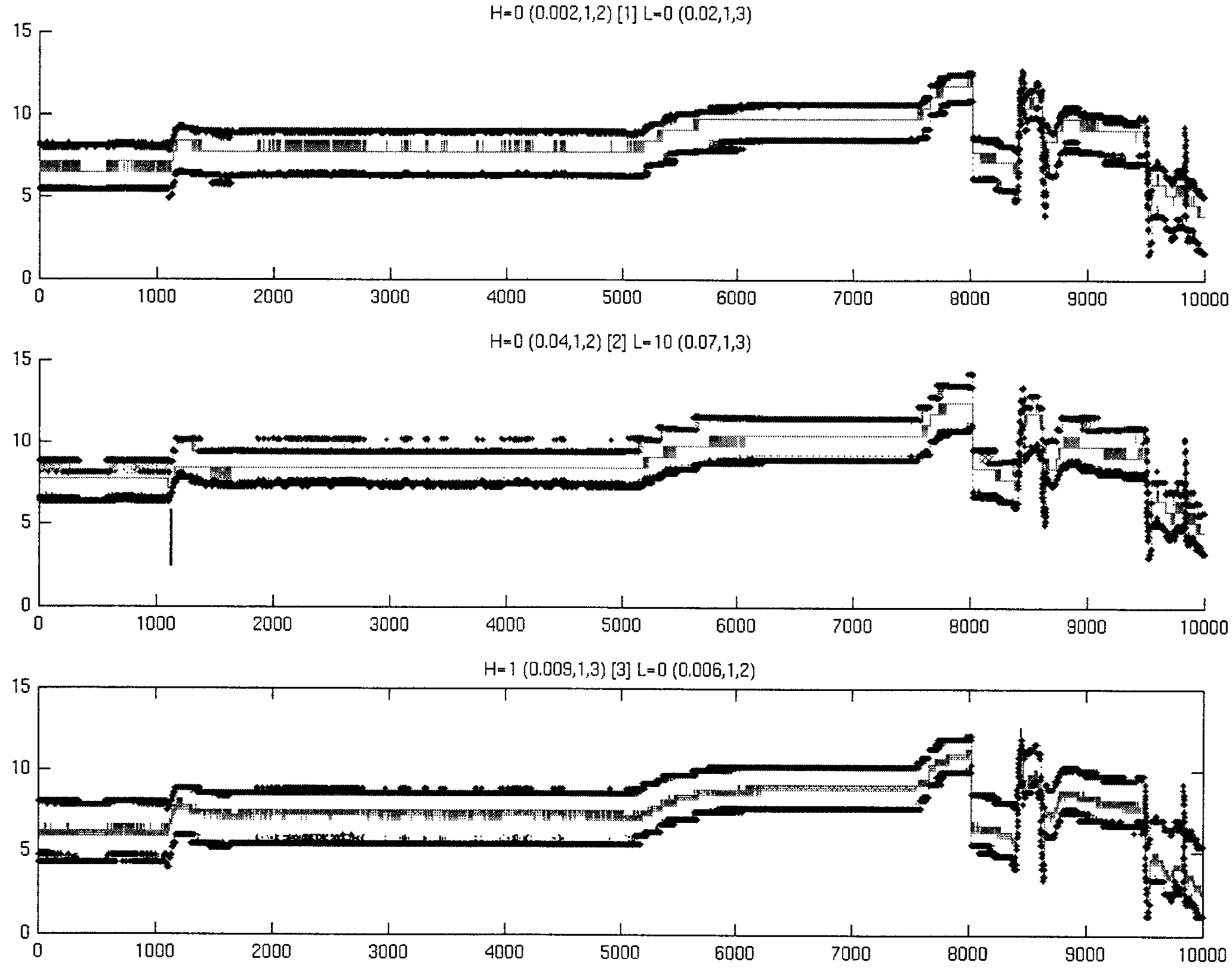

Figure 3 Training data for Sensors P-4011, P-4021, and P-4022

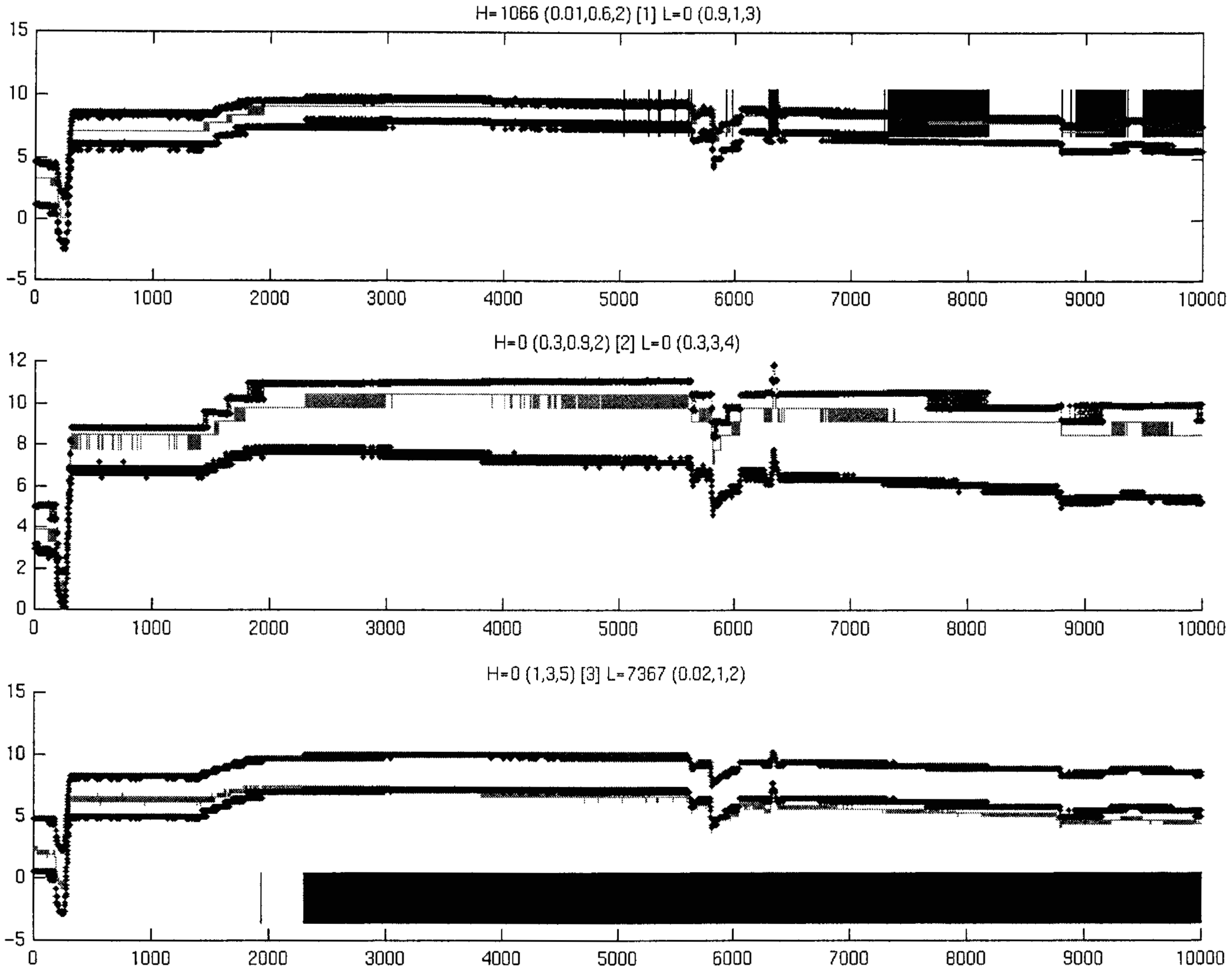

Figure 4 Test data for Sensors P-4011, P-4021, and P-4022

[2] Dennis DeCoste. “Mining multivariate time-series sensor data to discover behavior envelopes,” in Proceedings of the *Third Conference on Knowledge Discovery and Data Mining* (KDD-97), Newport Beach, CA, August 1997.

[3] Dennis DeCoste. “Automated learning and monitoring of limit functions,” in Proceedings of the *Fourth International Symposium on Artificial Intelligence, Robotics, and Automation for Space* (i-SAIRAS-97), Japan, July 1997.

[4] Dennis DeCoste. “Adaptive Resource Profiling,” in Proceedings of the *Fifth International Symposium on Artificial Intelligence, Robotics, and Automation for Space* (i-SAIRAS-99), The Netherlands, June, 1999.

[5] T. Estlin, S. Hayati, A. Jain, J. Yen, G. Rabideau, R. Castano, R. Petras, S. Peters, D. DeCoste, E. Tunstel, S. Chien, E. Mjolsness, R. Steele, D. Mutz, A. Gray, and T. Mann. “An Integrated Architecture for Cooperating Rovers,” in Proceedings of the *Fifth International Symposium on Artificial Intelligence and Automation in Space*, (iSAIRAS-99), Noordwijk, the Netherlands, June, 1999.

[6] David Nix and Andreas Weigend. “Learning Local Error Bars for Nonlinear Regression,”, NIPS-7, 1994.

[7] R.D. Reiss and M. Thomas. *Statistical Analysis of*

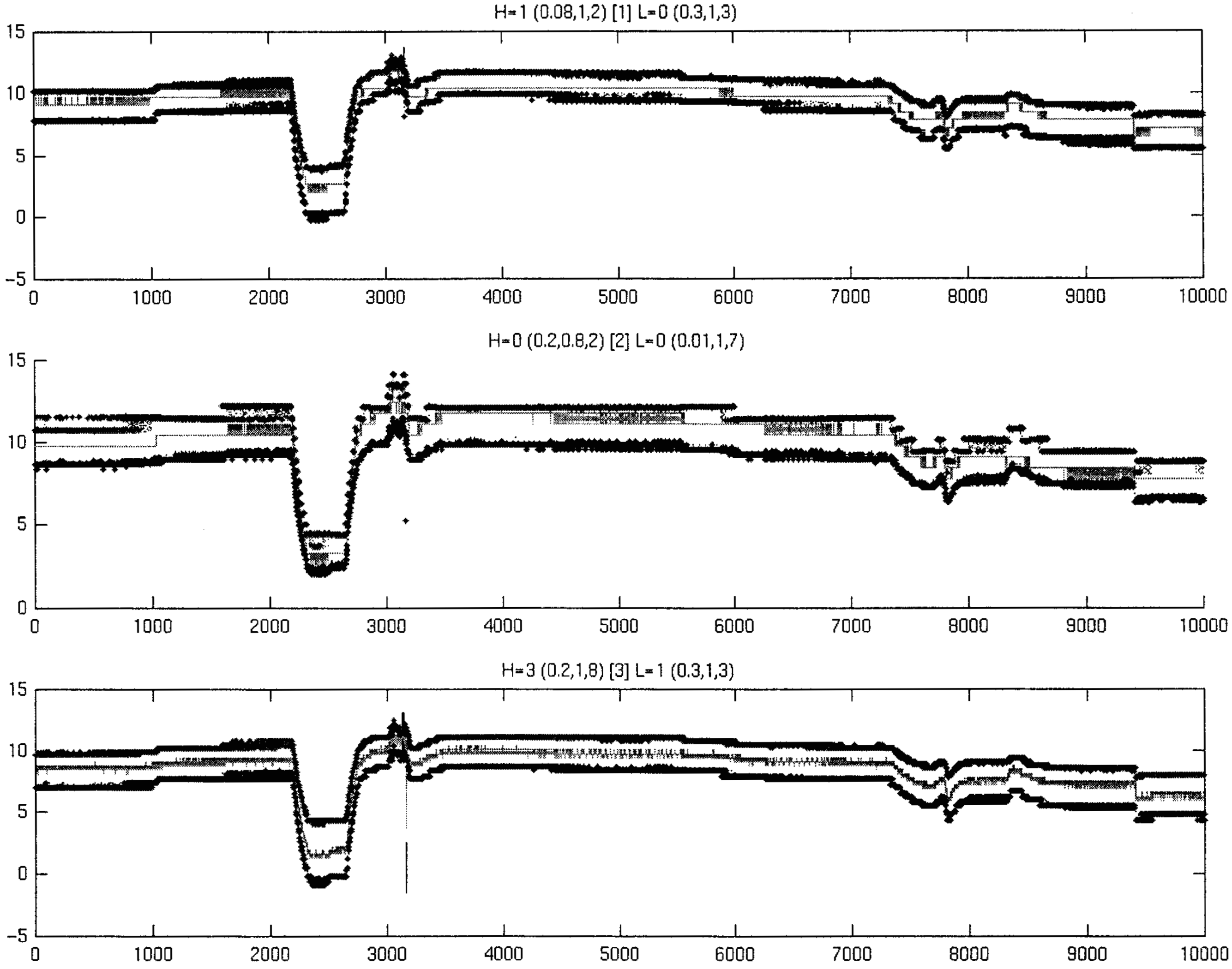

Figure 5 Second Test Data for Sensors P-4011, P-4021, and P-4022 (Occurred Before Training Data)

Extreme Values : From Insurance, Finance, Hydrology and Other Fields, Birkhäuser, 1997.

[8] Andeas Weigend and Ashok Srivastava. "Predicting Conditional Probability Distributions: A Connectionist Approach," *International Journal of Neural Systems*, Volume 6, 1995.

[9] C. J. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", *Data Mining and Knowledge Discovery*, Vol 2, Number 2, 1998. (also see http://svm.first.gmd.de).

Dennis DeCoste is Senior Member of Technical Staff and Technical Group Leader for the Machine Learning Systems Group at the Jet Propulsion Laboratory. He received his Ph.D. in Artificial Intelligence / Computer Science from the University of Illinois at Urbana-Champaign in 1994. He has been technical lead of several projects in fault detection and time-series data mining at JPL for the last five years. He has served on the AAAI program committee and as reviewer for AIJ, IEEE PAMI, AAAI, and IJCAI.